

Tipos de Función en JavaScript

Funciones declaradas

```
function nombre([parameters...]) {  
    ... instrucciones...  
    [return valor;]  
}
```



```
function cuadrado (numero) {  
    return numero * numero;  
}
```

*cuando las funciones son
anónimas se denominan también
funciones **lambda***

Funciones expresadas

```
variable = function nombre([parameters...]) {  
    ... instrucciones...  
    [return valor;]  
}
```

```
const sqr = function cuadrado (number)  
    { return number * number }
```

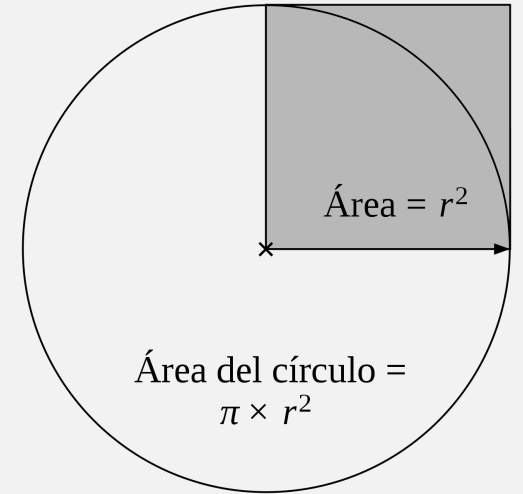
Este tipo de funciones pueden ser **anónimas**, es decir, no tienen por qué tener un nombre.

```
const sqr = function (number)  
    { return number * number }
```

Funciones expresadas

Este enfoque permite crear una función en el interior de una variable, lo que permitirá posteriormente ejecutar la variable (como si fuera una función).

```
const sqr = function (number) { return number * number }  
let radio = 2;  
alert(sqr(radio)); // muestra 4
```



Las funciones expresadas son convenientes cuando se pasa una función como argumento a otra función.

```
// MOSTRAR EL ÁREA DE UN CÍRCULO  
const sqr = function (number) { return number * number }  
function areaCirculo(radio2) {return Math.PI * radio2;}  
let radio = Number(prompt('Indica el área del círculo'));  
alert('El area del círculo es: ' + areaCirculo(sqr(radio)));
```

Funciones flecha

Las funciones flecha son una forma corta de escribir **funciones anónimas expresadas** a partir de **ECMAScript 6**. Para convertir una función en una función flecha...

1. Creamos la función

```
const saludar = function () {  
    alert("Hola gente");  
}
```

2. Quitamos la palabra function y entre el paréntesis y la llave escribimos => (lo cual forma una flecha)

```
const saludar = () => {  
    alert("Hola gente");  
}
```

3. Si la función tiene una sola instrucción se pueden omitir las llaves { } (de lo contrario no)

```
const saludar = () => alert("Hola gente");
```

4. Cuando una función flecha sólo recibe un parámetro se pueden eliminar los paréntesis.
(Si no hay parámetros o hay más de un parámetro, no.)
Ejemplo:

```
const saludo = persona => alert("Hola" + persona);
```

Y para llamarla...

```
saludar();  
// o  
saludo("Pep");
```